United States
Naval Postgraduate School

THESIS

IDENTIFICATION OF PLANT DYNAMICS
USING Ho's ALGORITHM

by

Ilker Eldem

June 1969

Library
Naval Postgraduate School
Monterey, Californi

Identification of Plant Dynamics

Using Ho's Algorithm

by

İlker Eldem
Lieutenant, Turkish Navy
Naval Academy, Turkey, 1961

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1969

ABSTRACT

Ho's Algorithm is reviewed and demonstrated with analytic examples. A digital computer program is developed to implement the algorithm for single-input, single-output systems and used to identify linear continuous and stationary systems which are driven with a unit step as the test input. Discrete realization of the continuous systems is obtained using the measured output-samples, to a step input, directly in the algorithm.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Doctor George J. Thaler of the Electrical Engineering Department of the Naval Postgraduate School for the guidance and helpful suggestions in the preparation of this work.

I. REVIEW OF Ho's ALGORITHM

A. INTRODUCTION

Design of a control system which will perform acceptably with respect to some criterion and satisfy the possible constraints over its operating range depends on knowledge of the process dynamics. The dynamics might be linear, nonlinear, time variant, stationary or might be a function of environment. In any case the dynamics of the process must be formulated by a set of differential equations (if it is a continuously operating system) or by a set of difference equations (if it is a discrete time system). An effective design and/or analysis can proceed after this formulation.

These equations might be completely known or might be written down from the parameter values supplied by the manufacturer using the laws of physics. Sometimes only partial information or no information is supplied about the process; yet somehow the dynamics must be formulated. This leads to the solution of an identification problem.

B. BACKGROUND

A number of algorithms can be found [1,2,3,5][1] for the solution of the identification problem. These range

---

[1] Numbers in brackets indicate the references at the end.

from a partial identification, like the damping factor of a dominant complex pair of poles, to a complete identification of the systems dynamics using frequency and time domain techniques.

The identification problem can be defined in general as: to find an internal description of the system (a set of differential or difference equations) from the given external description (input-output relation).

Input-output relations may be defined in the time domain (impulse response for continuous systems and pulse response for discrete systems) or in the frequency domain (transfer function). In the identification problem this relation is given in general as experimental data.

Most of the algorithms are applicable to linear, stationary systems only; and only this class of systems will be considered in this study.

The following definitions and notations will be used.

A continuous, linear, stationary system will be represented by

$$\begin{aligned} \dot{\underset{\sim}{x}} &= A\underset{\sim}{x} + B\underset{\sim}{u} \\ \underset{\sim}{y} &= C\underset{\sim}{x} \end{aligned} \qquad (1)[2]$$

where,

A is a nxn system matrix

B is a nxm distribution matrix

---

[2] $\dot{\underset{\sim}{x}}$ denotes $\frac{d}{dt}\underset{\sim}{x}$

C is a pxn observation matrix

$\underset{\sim}{x} = \underset{\sim}{x}(t)$ is a nxl state vector

$\underset{\sim}{u} = \underset{\sim}{u}(t)$ is a mxl input vector

$\underset{\sim}{y} = \underset{\sim}{y}(t)$ is a pxl output vector

and a discrete, linear, stationary system will be represented by

$$\begin{aligned} \underset{\sim}{x}(k+1) &= A_D \underset{\sim}{x}(k) + B_D \underset{\sim}{u}(k) \\ \underset{\sim}{y}(k) &= C_D \underset{\sim}{x}(k) \end{aligned} \qquad (2)$$

with dimensions same as (1)

A continuous system is said to be controllable [2] if the matrix

$$\left[B \;\vdots\; AB \;\vdots\; A^2B \;\vdots\; \cdots \;\vdots\; A^{n-1}B \right] \qquad (3)^3$$

has rank n; and is said to be observable if the matrix

$$\left[C^T \;\vdots\; A^TC^T \;\vdots\; (A^T)^2C^T \;\vdots\; \cdots \;\vdots\; (A^T)^{n-1}C^T \right] \qquad (4)^4$$

has rank n. Same definitions hold for a discrete system by replacing A, B, C with A_D, B_D, C_D respectively.

A system may always be partitioned into four possible subsystems [2,3] as shown in Fig. 1.

Part I: Controllable and observable

Part II: Uncontrollable but observable

Part III: Controllable but unobservable

Part IV: Uncontrollable and unobservable.

---

[3] Vertical dotted lines indicate partitioning.

[4] C^T indicates transpose of C.

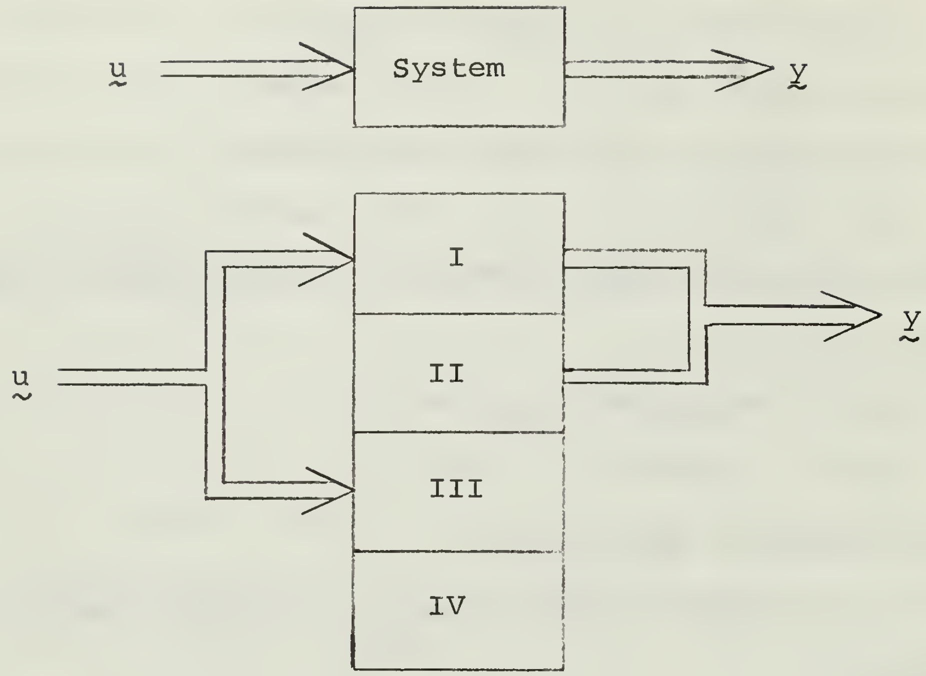

Fig. 1

As is seen from Fig. 1 only Part I gives the input-output relation. Although Part II seems to be contributing to the output, if no energy is stored at this part (zero initial condition) any contribution will be the result of a noise input only.

If the system is of the order n and the subsystems are of order n_I, n_{II}, n_{III}, n_{IV} respectively then

$$n = n_I + n_{II} + n_{III} + n_{IV} \tag{5}$$

Identification (alternatively called realization) of the system in Fig. 1 will give a system of order n_I.

The above statements are given as the following theorem in [2]. "Knowledge of the impulse response identifies the completely controllable and completely observable part, and this part alone, of the dynamical system which

generated it. This part is itself a dynamical system and has the smallest dimension among all realizations. Moreover, this part is identified by its impulse response uniquely up to algebraic equivalence".

Therefore complete identification of a dynamical system necessitates complete controllability and observability of the system. Part I must constitute all the system and parts II, III and IV must be missing.

In this study only completely controllable and observable systems will be considered.

C. Ho's ALGORITHM [1,4]

First the algorithm will be given for a discrete system as represented in (2).

Let $h(k)_{ij}$ be the present response at output i to a unit pulse at input j applied k time units ago. Let H_k be the matrix

$$H_k = \begin{bmatrix} h(k)_{11} & h(k)_{12} & \cdots & h(k)_{1m} \\ h(k)_{21} & h(k)_{22} & \cdots & \cdots \\ \vdots & & & \vdots \\ h(k)_{p1} & \cdots & \cdots & h(k)_{pm} \end{bmatrix} \tag{6}$$

Let HO_ℓ be the block matrix

$$HO_\ell = \begin{bmatrix} H_1 & H_2 & \cdots & H_\ell \\ H_2 & H_3 & \cdots & \cdot \\ \vdots & & & \vdots \\ H_\ell & \cdots & \cdots & H_{2\ell-1} \end{bmatrix} \tag{7}$$

Let ρ_ℓ be the rank of HO_ℓ. Build the block matrices HO_ℓ $\ell = 1,2,3,$ ... and each time find the rank ρ_ℓ, until ρ_ℓ equal $\rho_{\ell+1}$, let this value of ρ_ℓ be n and ℓ be q; then find the elementary transformation matrices P and Q such that

$$P(HO_q)Q = \begin{bmatrix} 1 & 0 & \cdots & & & & 0 \\ 0 & 1 & & & & & \vdots \\ \vdots & & & \overset{n\ 1\text{'s}}{1} & & & \vdots \\ \vdots & 0 & & & \ddots & & \vdots \\ \vdots & \vdots & & & 1 & & \vdots \\ \vdots & \vdots & & & & 0 & \vdots \\ \vdots & \vdots & & & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \tag{8}$$

Save P and Q matrices. Let HOT_q be the block matrix

$$HOT_q = \begin{bmatrix} H_2 & H_3 & \cdots & H_{q+1} \\ H_3 & H_4 & \cdots & \cdot \\ \vdots & \vdots & & \vdots \\ H_{q+1} & \cdots & & H_{2q} \end{bmatrix} \qquad (9)$$

Then a state variable description of the system is found by letting

$$\begin{aligned} A_D &= \text{the } n\text{x}n \text{ northwest corner of } P(HOT_q)Q \\ B_D &= \text{the } n\text{x}m \text{ northwest corner of } P(HO_q) \\ C_D &= \text{the } p\text{x}n \text{ northwest corner of } (HO_q)Q \end{aligned} \qquad (10)$$

The following results can be stated

1. $\rho_q = \rho_{q+1} = n$ is the minimum dimension for the realization.
2. By choosing P and Q suitably all possible realizations may be obtained.
3. Any two minimal realizations are isomorphic, that is there exists a nonsingular matrix W such that

$$\begin{aligned} A_{D_2} &= W\, A_{D_1}\, W^{-1} \\ B_{D_2} &= W\, B_{D_1} \\ C_{D_2} &= C_{D_1}\, W^{-1} \end{aligned} \qquad (11)$$

and W is given explicitly as

$$W = (V_2 V_2^T)^{-1} \; V_2 V_1^T \tag{12}$$

where V_r $r = 1,2$ is given by

$$V_r = \left[C_{D_r}^T \;\vdots\; A_{D_r}^T C^T \;\vdots\; (A_{D_r}^T)^2 C_{D_r}^T \;\vdots\; \cdots \;\vdots\; (A_{D_r}^T)^{q-1} C_{D_r}^T \right] \tag{13}$$

For continuous systems the algorithm follows the same pattern except in (6) each $h(k)_{ij}$ must be replaced by the impulse response and its successive derivatives evaluated at $t = 0$, that is, if $g(t)_{ij}$ is defined as the impulse response between input j and output i then

$$\begin{aligned} h(1)_{ij} &= g(0)_{ij} \\ h(2)_{ij} &= \dot{g}(0)_{ij} \\ h(3)_{ij} &= \ddot{g}(0)_{ij} \\ &\;\vdots \end{aligned} \tag{14}$$

or equally $g(t)_{ij}$ must be expanded in a power series for all t as

$$g(t)_{ij} = \sum_{k=1}^{\infty} a_k \; t^{k-1}/(k-1)! \tag{15}$$

and each $h(k)_{ij}$ must be replaced by a_k. These necessitate that $g(t)_{ij}$ must be a real analytic function.

If the input-output relation is given in the transform domain ($H(z)_{ij}$ for discrete systems and $H(s)_{ij}$ for continuous systems) it must be representable as a power series in the form of

$$H(z)_{ij} = \sum_{k=1}^{\infty} b_k \, z^{-k} \quad \text{(s for cont. systems)} \tag{16}$$

and each $h(k)_{ij}$ in (6) must be replaced by b_k, after that the algorithm is developed in similar fashion.

In any case H_k, $k = 1,2, \ldots$ constitutes a sequence of constant matrices. In a broad sense the identification problem may be restated as [1]: Given a sequence of pxm constant matrices, H_k, $k = 1,2, \ldots$ find the triple A_D, B_D, C_D (or A, B, C) of constant matrices such that

$$H_k = C_D A_D^{k-1} B_D \, , \quad k = 1,2, \ldots \tag{17}$$

to show the mechanics of the algorithm the following examples are presented.

1. <u>Example 1</u> Single-Input, Single-Output Discrete System ($m = 1$, $p = 1$)

Let the H_k $k = 1,2, \ldots$ be

$H_1 = 1 \quad H_2 = 0 \quad H_3 = -2$

$H_4 = 6 \quad H_5 = -14 \quad H_6 = 30 \ldots$

Start building HO_ℓ, $\ell = 1,2, \ldots$

$$HO_1 = \begin{bmatrix} H_1 \end{bmatrix} = \begin{bmatrix} 1 \end{bmatrix} \; ; \quad \rho_1 = 1$$

$$HO_2 = \begin{bmatrix} H_1 & H_2 \\ H_2 & H_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} ; \quad \rho_2 = 2$$

$$HO_3 = \begin{bmatrix} 1 & 0 & -2 \\ 0 & -2 & 6 \\ -2 & 6 & -14 \end{bmatrix} \quad ; \quad \rho_3 = 2$$

therefore

$$n = 2 \quad \text{and} \quad q = 2$$

$$HOT_2 = \begin{bmatrix} H_2 & H_3 \\ H_3 & H_4 \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -2 & 6 \end{bmatrix}$$

A set of P and Q matrices which satisfy (8) is

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}$$

$$P_1(HOT_2)Q_1 = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} = A_{D_1}$$

$$P_1(HO_2) = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} \Longrightarrow B_{D_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$(HO_2)Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \Longrightarrow C_{D_1} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

therefore the state equations are

$$\underset{\sim}{x}(k+1) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \underset{\sim}{x}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \underset{\sim}{x}(k)$$

this realization must satisfy (17)

$$H_k = C_{D_1} A_{D_1}^{k-1} B_{D_1} \quad , \quad k = 1, 2, \ldots$$

$A_{D_1}^{k-1}$ can be calculated by the Cayley-Hamilton technique:

$$A_{D_1}^{k-1} = \alpha_o I + \alpha_1 A_{D_1} = \begin{bmatrix} \alpha_o & \alpha_1 \\ -2\alpha_1 & \alpha_o - 3\alpha_1 \end{bmatrix}$$

$$H_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_o & \alpha_1 \\ -2\alpha_1 & \alpha_o - 3\alpha_1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \alpha_o$$

where α_o and α_1 can be found as follows:

Eigenvalues of A_{D_1} are

$$\lambda_1 = -1, \quad \lambda_2 = -2 \text{ ; therefore}$$

$$(\lambda_1)^{k-1} = (-1)^{k-1} = \alpha_o + \alpha_1 \lambda_1 = \alpha_o - \alpha_1$$

$$(\lambda_2)^{k-1} = (-2)^{k-1} = \alpha_o + \alpha_1 \lambda_2 = \alpha_o - 2\alpha_1$$

Substituting for k = 1,2, ...

$$H_1 = 2(-1)^{1-1} - (-2)^{1-1} = 1$$

$$H_2 = 2(-1)^{2-1} - (-2)^{2-1} = 0$$

$$H_3 = 2(-1)^{3-1} - (-2)^{3-1} = -2$$

$$\vdots \qquad \vdots \qquad \vdots$$

which are exactly the same as the starting H_k, k = 1,2, ...

A second set of P and Q matrices is

$$P_2 = \begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

which give the realization

$$\underset{\sim}{x}(k+1) = \begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix} \underset{\sim}{x}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \underset{\sim}{x}(k)$$

this realization also satisfies (17).

For the isomorphism of two realizations, W is calculated as:

$$W = (V_2 V_2^T)^{-1} V_2 V_1^T$$

By (13):

$$V_1 = \left[C_{D_1}^T \;\vdots\; A_{D_1}^T C_{D_1}^T\right] = \left[\begin{pmatrix}1\\0\end{pmatrix} \;\vdots\; \begin{pmatrix}0 & -2\\1 & -3\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\right] = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

$$V_2 = \left[\begin{pmatrix}1\\0\end{pmatrix} \;\vdots\; \begin{pmatrix}0 & 1\\-2 & -3\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\right] = \begin{bmatrix}1 & 0\\0 & -2\end{bmatrix}$$

$$W = \begin{bmatrix}1 & 0\\0 & -\frac{1}{2}\end{bmatrix} \quad ; \quad W^{-1} = \begin{bmatrix}1 & 0\\0 & -2\end{bmatrix}$$

$$A_{D_2} = WA_{D_1}W^{-1} = \begin{bmatrix}1 & 0\\0 & -\frac{1}{2}\end{bmatrix}\begin{bmatrix}0 & 1\\-2 & -3\end{bmatrix}\begin{bmatrix}1 & 0\\0 & -2\end{bmatrix} = \begin{bmatrix}0 & -2\\1 & -3\end{bmatrix}$$

$$B_{D_2} = WB_{D_1} = \begin{bmatrix}1 & 0\\0 & -\frac{1}{2}\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix}$$

$$C_{D_2} = C_{D_1}W^{-1} = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\0 & -2\end{bmatrix} = \begin{bmatrix}1 & 0\end{bmatrix}$$

The two realizations have the following signal flow graphs:

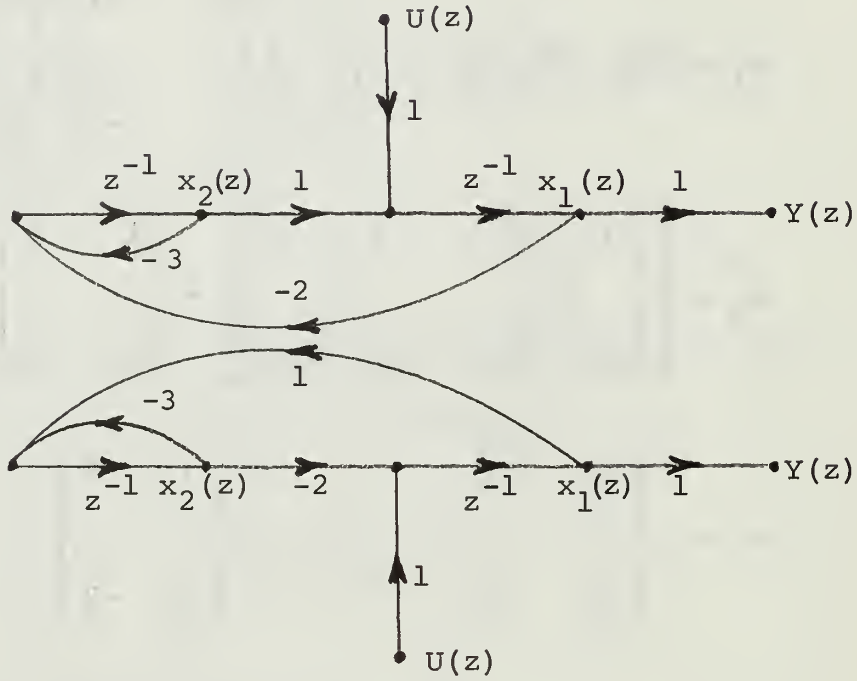

Fig. 2

By Mason's Gain Formula both have the transfer function

$$H(z) = \frac{(z+3)}{(z+1)(z+2)}$$

By a long division

$$H(z) = z^{-1} + 0z^{-2} - 2z^{-3} + 6z^{-4} - 14z^{-5} \ldots$$

the coefficients of z^{-k}, $k = 1,2, \ldots$ are again equal to H_k, $k = 1,2, \ldots$

2. Example 2 Two-Inputs , Three-Outputs

Continuous System ($m = 2$, $p = 3$)

Let the H_k, $k = 1,2, \ldots$ be

$$H_1 = \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 0 & 1 \\ -2 & -4 \\ -3 & -4 \end{bmatrix}, \quad H_3 = \begin{bmatrix} 1 & 1 \\ 4 & 8 \\ 8 & 14 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} -3 & -5 \\ -8 & -16 \\ -18 & -34 \end{bmatrix}, \quad H_5 = \begin{bmatrix} 7 & 13 \\ 16 & 32 \\ 38 & 74 \end{bmatrix}$$

Then

$$HO_1 = \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad \rho_1 = 2$$

$$HO_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & -2 & -4 \\ 2 & 1 & -3 & -4 \\ 0 & 1 & 1 & 1 \\ -2 & -4 & 4 & 8 \\ -3 & -4 & 8 & 14 \end{bmatrix} \quad \rho_2 = 3$$

$$HO_3 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 2 & -2 & -4 & 4 & 8 \\ 2 & 1 & -3 & -4 & 8 & 14 \\ 0 & 1 & 1 & 1 & -3 & -5 \\ -2 & -4 & 4 & 8 & -8 & -16 \\ -3 & -4 & 8 & 14 & -18 & -34 \\ 1 & 1 & -3 & -5 & 7 & 13 \\ 4 & 8 & -8 & -16 & 16 & 32 \\ 8 & 14 & -18 & -34 & 38 & 74 \end{bmatrix} \quad \rho_3 = 3$$

therefore

$n = 3$ and $q = 2$

$$HOT_2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ -2 & -4 & 4 & 8 \\ -3 & -4 & 8 & 14 \\ 1 & 1 & -3 & -5 \\ 4 & 8 & -8 & -16 \\ 8 & 14 & -18 & -34 \end{bmatrix}$$

A set of P and Q matrices is

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -2 & 0 & 1 & -1 & 0 & 0 \\ 2 & 1 & -2 & 0 & 1 & -1 \\ -2 & 0 & 2 & 2 & 1 & 0 \\ -1 & 0 & 1 & -1 & -1 & 1 \end{bmatrix}$$

$$Q_1 = \begin{bmatrix} 1 & 0 & 3 & -4 \\ 0 & 1 & -2 & 3 \\ 0 & 0 & 5 & -7 \\ 0 & 0 & -3 & 4 \end{bmatrix}$$

which give the realization

$$\dot{\underset{\sim}{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ -4 & -7 & -4 \end{bmatrix} \underset{\sim}{x} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \underset{\sim}{u}$$

$$\underset{\sim}{y} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \underset{\sim}{x}$$

this has the impulse response matrix

$$G(t) = \begin{bmatrix} 3/2 - \epsilon^{-t} + \frac{1}{2}\epsilon^{-2t} & 2-3\epsilon^{-t} + 2\epsilon^{-2t} \\ \epsilon^{-2t} & 2\epsilon^{-2t} \\ 3/2 - 2\epsilon^{-t} + 5/2\epsilon^{-2t} & 2-6\epsilon^{-t} + 5\epsilon^{-2t} \end{bmatrix}$$

Expanding the exponentials in power series and expressing the result as a matrix polynomial the following expression can be obtained.

$$G(t) = \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -2 & -4 \\ -3 & -4 \end{bmatrix} t + \begin{bmatrix} 1 & 1 \\ 4 & 8 \\ 8 & 14 \end{bmatrix} t^2/2!$$

$$+ \begin{bmatrix} -3 & -5 \\ -8 & -16 \\ -18 & -34 \end{bmatrix} t^3/3! + \begin{bmatrix} 7 & 13 \\ 16 & 32 \\ 38 & 74 \end{bmatrix} t^4/4! + \cdots$$

Each element of the coefficient matrices corresponds to the a_k's of (15) and the matrices themselves are exactly H_k's.

A second set of P and Q matrices is

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & -2 & 0 & 1 & -1 \\ -2 & 0 & 2 & 2 & 1 & 0 \\ -1 & 0 & 1 & -1 & -1 & 1 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} -2 & -3 & 3 & -4 \\ 2 & 3 & -2 & 3 \\ -5 & -5 & 5 & -7 \\ 3 & 3 & -3 & 4 \end{bmatrix}$$

which give the realization

$$\dot{\underset{\sim}{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix} \underset{\sim}{x} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \underset{\sim}{u}$$

$$\underset{\sim}{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \underset{\sim}{x}$$

It can be shown as before that this realization also satisfies (17).

To show the isomorphism of two realizations W is calculated as

$$V_1 = \left[C_1^T \;\vdots\; A_1^T C_1^T \right]$$

$$= \begin{bmatrix} 1 & 1 & 2 & 0 & -2 & -3 \\ 0 & 2 & 1 & 1 & -4 & -4 \\ 0 & 1 & 1 & 0 & -2 & -3 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & -2 & -1 \\ 0 & 1 & 1 & 0 & -2 & -3 \end{bmatrix}$$

$$(V_2 V_2^T)^{-1} = \frac{1}{75} \begin{bmatrix} 41 & 8 & -7 \\ 8 & 29 & -16 \\ -7 & -16 & 14 \end{bmatrix}$$

$$V_2 V_1^T = \begin{bmatrix} 3 & 1 & 1 \\ 8 & 15 & 8 \\ 16 & 23 & 15 \end{bmatrix}$$

$$W = (V_2V_2^T)^{-1} V_2V_1^T$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$W^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}$$

$$A_2 = WA_1W^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ -4 & -7 & -4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ -1 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix}$$

$$B_2 = WB_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$C_2 = C_1W^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

The W matrix also relates the two sets of P and Q matrices as

$$P_2 = \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} P_1 \quad , \quad Q_2 = Q_1 \begin{bmatrix} W^{-1} & 0 \\ 0 & I \end{bmatrix}$$

For this specific example

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} P_1$$

$$Q_2 = Q_1 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

D. ANOTHER ALGORITHM

The algorithm given in [5] for discrete systems (equally applicable to continuous systems) with scalar measurement (single-output) shows a similarity with Ho's Algorithm.

For the algorithm, a free dynamical system (no forcing function) represented as in (18) is considered

$$\begin{aligned} \underset{\sim}{x}(k+1) &= A_D\, \underset{\sim}{x}(k) \\ \underset{\sim}{y}(k) &= C_D\, \underset{\sim}{x}(k) \end{aligned} \tag{18}$$

where

A_D is a nxn matrix

C_D is a 1xn row vector

x is a nx1 state vector

y is the scalar output

and x(0) is known. Then HO_ℓ and HOT_ℓ matrices are constructed as in Ho's Algorithm for single-input, single-output systems, from the measured output-sequence. An algebraic equivalent of the system is realized by letting

$$\begin{aligned} A_D' &= (HOT_q)(HO_q)^{-1} \\ C_D' &= [1 \quad 0 \quad \cdots\cdots \quad 0] \end{aligned} \tag{19}$$

Note that for single-input, single-output systems of order n, q is equal to n, and ρ also is equal to n. Then this algorithm becomes a special case of Ho's Algorithm by letting

$$\begin{aligned} P &= I \\ Q &= (HO_q)^{-1} \end{aligned} \tag{20}$$

Because a nonsingular matrix can always be reduced to an identity matrix by elementary transformations on its columns (or rows) only.

For complete identification of the system, complete controllability and observability was a necessary and sufficient condition for systems with forcing. Here, as B_D is absent, controllability cannot be checked; instead a new condition called n-identification is imposed.

A system is said to be n-identifiable if the matrix

$$\left[\underset{\approx}{x}(0) \;\vdots\; A_D\underset{\approx}{x}(0) \;\vdots\; A_D^{\,2}\underset{\approx}{x}(0) \;\vdots\; \cdots \;\vdots\; A_D^{\,n-1}\underset{\approx}{x}(0)\right] \tag{21}$$

has rank n. Similarity of (3) and (21) is obvious for single-input systems.

Therefore it is possible to determine A_D' and C_D' if the system is observable and n-identifiable. These two conditions are summed together and called 1-identifiability [5].

In fact another realization is possible by letting

$$\begin{aligned} A_D'' &= (HO_q)^{-1}\,(HOT_q) \\ C_D'' &= [\text{first row of } HO_q] \end{aligned} \tag{22}$$

This is also a special case of Ho's Algorithm; simply

$$\begin{aligned} P &= (HO_q)^{-1} \\ Q &= I \end{aligned} \tag{23}$$

The following relation

$$A_D'' = (A_D')^T \tag{24}$$

holds for these two extreme cases. As HO and HOT are symmetric matrices for single-input,single-output systems it can be written

$$(A_D')^T = \left[(HOT_q)(HO_q)^{-1}\right]^T$$

$$= \left[(HO_q)^{-1}\right]^T \left[HOT_q\right]^T \tag{25}$$

$$= (HO_q)^{-1}(HOT_q) = A_D''$$

Example 1 is applicable to these cases just simply letting

$$\underset{\sim}{x}(0) = B_D$$

II. APPLICATION OF Ho's ALGORITHM

A. IMPLEMENTATION OF Ho's ALGORITHM

Ho's Algorithm perhaps is the one which requires minimum computation among the existing identification schemes. Because of the simplicity of the computations it can be used in many applications.

The algorithm is implemented with a digital computer program to see how it can be used in an off line fashion for identification. The results obtained can be extended for use in other applications.

1. Digital Computer Programs[5]

The program consists of one main program and four subroutines. The main program reads the data, builds the HO and HOT matrices and controls the subroutines. For digital simulation of systems, data can be generated within the main program by replacing the data reading loop with a generating function. The algorithm does not stop when two successive ranks of HO are equal, but uses all available data for additional checking.

Subroutine SDWFD2 finds the successive derivatives evaluated at zero time for continuous systems; for discrete systems it is by-passed. FLAG controls this operation.

---

[5] See pages 56-61

Four Newton Forward Differences are used to evaluate each derivative. Subroutine RANK finds the rank of HO each time it is constructed. Subroutine PHOQ finds the elementary transformation matrices P and Q which put HO into its normal form. Subroutine MULT makes the matrix multiplications required by the algorithm.

The program is written for single input-output systems, but can be altered for multiple input-output systems; in fact only the reading data and building HO and HOT parts need changes in the main program. No change is required in the subroutines.

B. APPLICATION TO CONTINUOUS SYSTEMS

For discrete systems the algorithm is straight forward, the only limiting factor is the accuracy of the measurements. For continuous systems however some difficulties arise. First, as a test input signal an impulse is needed. For linear systems the impulse response is the derivative of the response to a step input. Therefore a step input can be used as the test input. Higher order inputs can also be used if it is necessary as long as the right derivatives are obtained. The program is written to accept the step response of the system.

A second problem comes from the numerical evaluation of successive derivatives from the measured samples. Numerical evaluation of a derivative is an approximation even with exact measurements, when higher derivatives are

needed accuracy diminishes greatly. A small error in the input data deteriorates the results, as the numerical methods magnify these errors enormously. The sampling period is the biggest factor in this magnification as it appears in the denominator, and for each additional higher derivative its power is increased by one. Magnification of error increases as sampling period decreases; but on the other hand for fast changing functions a small sampling period is needed.

The examples presented in the following tables were simulated digitally for unit step-inputs to the systems. Two parameters, DX and DD, in addition to the sampling period, seemed to have great effect on the accuracy of the obtained results. A suitable choice of the three gave repeating answers when the dimension of the HO matrix became larger than the order of the system. DX and DD are positive small numbers; when finding the rank of HO, any number which has an absolute value less than DX is equated to zero by the program. When finding the P and Q matrices, which put HO to its normal form, any number which has an absolute value less than DD is made equal to zero by the program. The values of these parameters which gave a good result for this specific example are included in the tables.

In example 3 a system with two real poles is considered. The poles are not greatly separated and a good realization became possible because the step response is a

fairly smooth function, and in the evaluation of the derivatives not too much error is involved, in fact the first four derivatives are almost exact. In Example 4 a system with three real poles, which are separated by a significant amount, is considered. In part A, the sampling rate is not so high. Identification of the pole at the origin is almost exact, the pole at -1000 is identified fairly well but the far pole is in error by a large amount. In part B, the sampling rate is increased; while the pole at -1000 is identified more accurately and the far pole identification showed improvement, identification of the pole at the origin began to deteriorate. In Example 5, a highly damped complex pair of poles gave a very accurate identification with a high sampling rate; as there is no spread of poles a good choice of sampling rate results in an accurate realization. In Example 6, a very lightly damped complex pole pair is considered. Again identification is very accurate because of the same reasons in Example 5.

In the above examples simulation of systems was carried out digitally in double-precision. In general a high sampling rate with a high-precision measurement is necessary which makes the scheme impractical to use with measurement of analog signals. A different approach is necessary which is the subject of the next section.

EXAMPLE 3			
Transfer Function of The System	$\frac{300}{(s+5)(s+20)}$		
Sampling Per.	0.005		
DX	0.005		
DD	0.000001		
Numerically Evaluated Successive Derivatives	0.3376×10^{-3}	0.3000×10^{3}	-0.7500×10^{4}
	0.1575×10^{6}	-0.3187×10^{7}	0.6393×10^{8}
	-0.1278×10^{10}	0.2448×10^{11}	0.2830×10^{13}
Realized System Matrix	-0.2132×10^{-13}	-0.1000×10^{3}	
	0.1000×10^{1}	-0.2500×10^{2}	
Realized Distribution Matrix ^{T}	0.1000×10^{1}	0.2541×10^{-20}	
Realized Observation Matrix	0.3376×10^{-3}	0.3000×10^{3}	
Eigenvalues of System Matrix	$-0.4999 \times 10^{1} + j\,0.0$		
	$-0.1999 \times 10^{2} + j\,0.0$		

EXAMPLE 4a			
Transfer Function of The System	$\dfrac{10^{12}}{s(s+1000)(s+1000000)}$		
Sampling Per.	0.001		
DX	1.0		
DD	1.0/|largest element of HO| x 2.0		
Numerically Evaluated Successive Derivatives	0.430×10^{2}	0.9148×10^{6}	0.8746×10^{9}
	0.8361×10^{12}	-0.7993×10^{15}	0.7641×10^{18}
	-0.7304×10^{21}	0.6981×10^{24}	-0.6670×10^{27}
Realized System Matrix	-0.5684×10^{-13}	-0.1309×10^{-9}	0.2131×10^{-1}
	0.1000×10^{1}	0.3725×10^{-8}	-0.1991×10^{7}
	0.1065×10^{-13}	0.1000×10^{1}	-0.3039×10^{4}
Realized Distribution Matrix T	0.1000×10^{1}	0.3552×10^{-14}	-0.1387×10^{-16}
Realized Observation Matrix	0.430×10^{2}	0.9148×10^{6}	-0.8746×10^{9}
Eigenvalues of System Matrix	$-0.2083 \times 10^{4} + j\,0.0$		
	$-0.956 \times 10^{3} + j\,0.0$		
	$0.1070 \times 10^{-7} + j\,0.0$		

EXAMPLE 4b	
Transfer Function of The System	$\dfrac{10^{12}}{s(s+1000)(s+1000000)}$
Sampling Per.	0.0001
DX	1.0
DD	1.0/|Largest element of HO| x2.0
Numerically Evaluated Successive Derivatives	-0.9625 $0.1000\text{x}10^{7}$ $-0.9918\text{x}10^{9}$ $0.8123\text{x}10^{12}$ $0.2927\text{x}10^{16}$ $-0.8084\text{x}10^{20}$ $0.1704\text{x}10^{25}$ $-0.3552\text{x}10^{29}$ $0.7400\text{x}10^{33}$
Realized System Matrix	$0.8633\text{x}10^{-12}$ $-0.5544\text{x}10^{-8}$ $-0.5187\text{x}10^{3}$ $0.1000\text{x}10^{1}$ $-0.1989\text{x}10^{-11}$ $-0.2083\text{x}10^{8}$ $0.6098\text{x}10^{-19}$ $0.1000\text{x}10^{1}$ $-0.2183\text{x}10^{5}$
Realized Distribution Matrix T	$0.1000\text{x}10^{1}$ $-0.5963\text{x}10^{-18}$ $-0.2646\text{x}10^{-22}$
Realized Observation Matrix	-0.9625 $0.1000\text{x}10^{7}$ $-0.9918\text{x}10^{9}$
Eigenvalues of System Matrix	$-0.2083 \times 10^{5} + j\ 0.0$ $-0.9999 \times 10^{3} + j\ 0.0$ $-0.2490 \times 10^{-4} + j\ 0.0$

EXAMPLE 5			
Transfer Function of The System	$\dfrac{1000000}{(s+25000 + j25000)(s+25000 - j25000)}$		
Sampling Per.	0.000001		
DX	0.000005		
DD	1.0/| Largest element of HO | x 2.0		
Numerically Evaluated Successive Derivatives	$0.6764\text{x}10^{1}$	$0.8164\text{x}10^{6}$	$-0.4927\text{x}10^{11}$
	$0.1443\text{x}10^{6}$	$-0.1056\text{x}10^{20}$	$-0.1306\text{x}10^{25}$
	$0.3408\text{x}10^{3}$	$-0.2533\text{x}10^{37}$	$0.2595\text{x}10^{44}$
Realized System Matrix	$0.9094\text{x}10^{-11}$	$-0.1250\text{x}10^{10}$	
	$0.1000\text{x}10^{1}$	$-0.5000\text{x}10^{5}$	
Realized Distribution Matrix^T	$0.1000\text{x}10^{1}$	$0.1058\text{x}10^{-20}$	
Realized Observation Matrix	$0.6764\text{x}10^{1}$	$0.8164\text{x}10^{6}$	
Eigenvalues of System Matrix	$-0.2500 \times 10^{5} + j\ 0.2500 \times 10^{5}$		
	$-0.2500 \times 10^{5} - j\ 0.2500 \times 10^{5}$		

EXAMPLE 6			
Transfer Function of The System	$\frac{1001}{(s + 0.01 + j500)(s + 0.01 - j500)}$		
Sampling Per.	0.0001		
DX	0.005		
DD	0.000001		
Numerically Evaluated Successive Derivatives	$-0.2084\text{x}10^{-6}$	$-0.1001\text{x}10^{4}$	$-0.1986\text{x}10^{2}$
	$-0.2502\text{x}10^{9}$	$0.1128\text{x}10^{8}$	$0.6234\text{x}10^{14}$
	$0.2890\text{x}10^{17}$	$-0.3539\text{x}10^{22}$	$0.4010\text{x}10^{27}$
Realized System Matrix	$0.8198\text{x}10^{-13}$	$-0.2500\text{x}10^{6}$	
	$0.1000\text{x}10^{1}$	$-0.1984\text{x}10^{-1}$	
Realized Distribution Matrix ^{T}	$0.1000\text{x}10^{1}$	$-0.2082\text{x}10^{-9}$	
Realized Observation Matrix	$-0.2084\text{x}10^{-6}$	$0.1001\text{x}10^{4}$	
Eigenvalues of System Matrix	$-0.9922 \times 10^{-2} + j\ 0.4999 \times 10^{3}$		
	$-0.9922 \times 10^{-2} - j\ 0.4999 \times 10^{3}$		

C. DISCRETE REALIZATION OF CONTINUOUS SYSTEMS

As is seen from the preceding examples, to find a set of differential equations for continuous systems is somewhat a trial and error procedure, as one must come up with a suitable triple of DX, DD and sampling period, if no apriori knowledge is available for the system. More importantly, very accurate measurements are needed which is impossible in an actual experiment with hardware.

The main error source in the above procedure is the evaluation of derivatives numerically. If the measured data can be used directly, without evaluating derivatives, inclusion of this error to the algorithm can be eliminated.

This approach leads to a system which is operating continuously but observed discretely as in Fig. 3

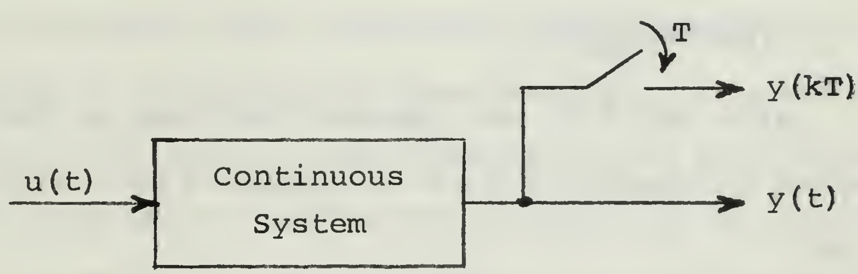

Fig. 3

where T is the sampling period and u(t) is a step. If the system has the dynamics as in (26)

$$\begin{aligned} \dot{\underset{\sim}{x}} &= A\underset{\sim}{x} + Bu \\ y &= C\underset{\sim}{x} \end{aligned} \qquad (26)$$

Then

$$\underset{\sim}{x}(t) = \mathcal{L}^{-1}\left\{(sI-A)^{-1}\underset{\sim}{x}(t_o) + (sI-A)^{-1}Bu(s)\right\} \tag{27}$$

As u(t) is a step (amplitude V)

$$\underset{\sim}{x}(t) = \mathcal{L}^{-1}\left\{(sI-A)^{-1}\underset{\sim}{x}(t_o) + \frac{(sI-A)^{-1}B}{s}V\right\} \tag{28}$$

Defining

$$\begin{aligned} \phi(t-t_o) &\triangleq \mathcal{L}^{-1}\left\{(sI-A)^{-1}\right\} \\ \Gamma(t-t_o) &\triangleq \mathcal{L}^{-1}\left\{\frac{(sI-A)^{-1}B}{s}\right\} \end{aligned} \tag{29}$$

Substituting in

$$\underset{\sim}{x}(t) = \phi(t-t_o)\underset{\sim}{x}(t_o) + \Gamma(t-t_o)V \tag{30}$$

letting

$$\begin{aligned} t_o &= kT \\ t &= t_o + T \end{aligned} \tag{31}$$

$$\underset{\sim}{x}[(k+1)T] = \phi(T)\underset{\sim}{x}(kT) + \Gamma(T)V \tag{32}$$

and

$$y(kT) = C\underset{\sim}{x}(kT) \tag{33}$$

$\phi(T)$ and $\Gamma(T)$ are constant matrices as the sampling period is constant, V can be replaced by v(kT) in (32) letting

$$v(kT) = 1 \quad , \quad k = 0,1,2, \ldots \tag{34}$$

as the test input is a unit step.

Dynamics described by (32) and (33) can be represented as a discrete system of Fig. 4.

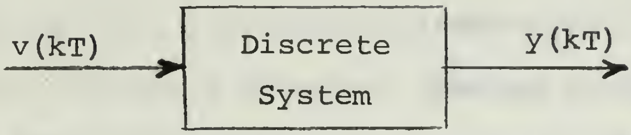

Fig. 4

If Fig. 3 is driven by a unit step and Fig. 4 with a unit impulse train of period T, they both will give the same y(kT). Therefore samples taken as the result of experiment on Fig. 3 can be used to identify the system in Fig. 4 as if it is driven by a unit impulse train and the output is measured.

Dynamics described in (32) and (33) is an exact representation of the continuous system of Fig. 3 at sampling instants and constitutes a discrete realization of the continuous system. Two points of importance must be mentioned here. One is the loss of controllability and observability due to the sampling of continuous systems [2], as a result a realization which is not a faithful representation of the system will be obtained. If a linear-stationary-continuous system is controllable and observable, it will remain so when sampling is introduced if and only if [6]

$$\operatorname{Re}[s_i] = \operatorname{Re}[s_j] \text{ implies } \operatorname{Im}[s_i - s_j] \neq \frac{k2\pi}{T} \tag{35}$$

where

s_m , m = 1,2, ... n eigenvalues of system

i , j = 1,2, ... n

k = positive integer

T = sampling period

If no apriori knowledge is available about the system, at least two realizations must be obtained with different sampling rates to avoid misrepresentation.

The second point comes from the usage of a unit step as the test input. As a result the discrete equivalent in Fig. 4 is driven with a unit impulse train. If the original system has a transfer function

$$H(z) = \frac{N(z)}{D(z)} = a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \cdots \tag{36}$$

in the z-domain, the realization will be for

$$H'(z) = \frac{zN(z)}{(z-1)D(z)} = a_1' z^{-1} + a_2' z^{-2} + a_3' z^{-3} + \cdots \tag{37}$$

a_i' , i = 1,2, ... are measured sample values to a unit step input. This will give an additional eigenvalue increasing the order of the system by one. This can be avoided by a simple manipulation of the data.

$$H(z) = H'(z)(1-z^{-1}) = a_1' z^{-1} + (a_2' - a_1') z^{-2} + \cdots \tag{38}$$

Therefore

$$\begin{aligned} a_1 &= a_1' \\ a_2 &= a_2' - a_1' \\ a_3 &= a_3' - a_2' \end{aligned} \tag{39}$$

This manipulation of the data has an additional benefit, if there is an unknown biasing in the measurement it is removed automatically except from the first sample. The program takes care of this conversion of the data.

In the following examples systems are simulated digitally. To see the sensitivity of the algorithm to numerical precision, sample values were rounded-off to four significant digits, and a realization without manipulation of the data is included in some examples to make comparisons possible. As will be seen the choices of DX and DD are not critical as in the continuous representation of systems.

Example 7a shows the identification of the additional pole at the origin because of a unit step test signal. In 7b this pole is removed with the manipulation of the data, and six significant digits are used for sample values, realization is very accurate. In 7c sample values are rounded to four significant digits, which results in a deterioration in the answer. In Example 8a and 8b seven significant digits are used. In 8a an additional pole is present because of step input, in 8b it is removed; with the same significant digits 8b gave a more accurate answer as the system order decreased by one, less computation is required and error is decreased. In 8c four significant digits are used with some deviation of the answer as a result. In Example 9a and 9b a complex pole pair and a real

pole are considered, with eight and four significant digits of sample values respectively. Highly accurate results are obtained in both cases. Example 10a shows the loss of controllability and observability (as the sampling is introduced in the measurement, only observability is lost) due to the sampling. In 10b with a different sampling rate the system is identified completely.

EXAMPLE 7a			
Transfer Function of The System	$\frac{160}{(s+1)(s+2)}$		
Sampling Per.	0.1		
DX	0.001		
DD	0.00001		
The Type of The Data	Not manipulated to take the step out Double-Precision		
Realized System Matrix	0.3552×10^{-14}	0.0	0.7408
	0.1000×10^{1}	0.3552×10^{-14}	-0.2464×10^{1}
	0.6661×10^{-15}	0.1000×10^{1}	0.2723×10^{1}
Realized Distribution Matrix ^{T}	0.1000×10^{1}	0.3552×10^{-14}	-0.2220×10^{-15}
Realized Observation Matrix	0.7244	0.2628×10^{1}	0.5374×10^{1}
Eigenvalues of System Matrix	$0.8187 + j\ 0.1230 \times 10^{-14}$ $0.9048 - j\ 0.7288 \times 10^{-15}$ $0.1000 \times 10^{1} - j\ 0.2778 \times 10^{-15}$		
Corresponding s-Domain Eigenvalues	-2.0 -1.0 0.0		

EXAMPLE 7b		
Transfer Function of The System	$\frac{160}{(s+1)(s+2)}$	
Sampling Per.	0.1	
DX	0.001	
DD	0.00001	
The Type of The Data	Manipulated to take the step out Rounded to six significant digits	
Realized System Matrix	0.2220×10^{-15} 0.1000×10^{1}	-0.7408 0.1723×10^{1}
Realized Distribution Matrix ^{T}	0.1000×10^{1}	-0.5551×10^{-16}
Realized Observation Matrix	0.7244	0.1904×10^{1}
Eigenvalues of System Matrix	0.8187 + j 0.0 0.9048 + j 0.0	
Corresponding s-Domain Eigenvalues	-2.0 -1.0	

EXAMPLE 7c		
Transfer Function of The System	$\frac{160}{(s+1)(s+2)}$	
Sampling Per.	0.1	
DX	0.001	
DD	0.00001	
The Type of The Data	Manipulated to take the step out Rounded to four significant digits	
Realized System Matrix	0.6661×10^{-15} 0.1000×10^{1}	-0.7386 0.1722×10^{1}
Realized Distribution Matrix^T	0.1000×10^{1}	-0.6938×10^{-16}
Realized Observation Matrix	0.7245	0.1904×10^{1}
Eigenvalues of System Matrix	$0.8069 + j\ 0.0$ $0.9154 + j\ 0.0$	
Corresponding s-Domain Eigenvalues	-2.145 -0.885	

EXAMPLE 8a				
Transfer Function of The System	$\frac{4000}{(s+1)(s+5)(s+10)}$			
Sampling Per.	0.1			
DX	0.001			
DD	0.00001			
The Type of The Data	Not manipulated to take the step out Rounded to seven significant digits			
Realized System Matrix	0.426×10^{-13}	-0.5684×10^{-13}	-0.5684×10^{-13}	-0.2030
	0.1000×10^{1}	-0.5684×10^{-13}	-0.5684×10^{-13}	0.1310×10^{1}
	0.0	0.1000×10^{1}	0.5684×10^{-13}	-0.2987×10^{1}
	-0.3907×10^{-13}	0.0	0.1000×10^{1}	0.2880×10^{1}
Realized Distribution Matrix ^{T}	0.1000×10^{1}	0.0	0.0	0.0
Realized Observation Matrix	0.4537	0.2542×10^{1}	0.6169×10^{1}	0.1077×10^{2}
Eigenvalues of System Matrix	$0.9987 - j\,0.4198\times10^{-14}$			
	$0.9073 - j\,0.2743\times10^{-13}$			
	$0.3715 - j\,0.2742\times10^{-13}$			
	$0.6028 + j\,0.5936\times10^{-13}$			
Corresponding s-Domain Eigenvalues	0.0			
	-0.972			
	-9.9			
	-5.06			

EXAMPLE 8b			
Transfer Function of The System	$\frac{4000}{(s+1)(s+5)(s+10)}$		
Sampling Per.	0.1		
DX	0.001		
DD	0.00001		
The Type of The Data	Manipulated to take the step out Rounded to seven significant digits		
Realized System Matrix	$0.2220\text{x}10^{-15}$	$-0.1554\text{x}10^{-14}$	0.2018
	$0.1000\text{x}10^{1}$	$0.3552\text{x}10^{-14}$	$-0.1104\text{x}10^{1}$
	$0.2220\text{x}10^{-15}$	$0.1000\text{x}10^{1}$	$0.1879\text{x}10^{1}$
Realized Distribution Matrix ^{T}	$0.1000\text{x}10^{1}$	$0.1554\text{x}10^{-14}$	$-0.4440\text{x}10^{-15}$
Realized Observation Matrix	0.4537	$0.2088\text{x}10^{1}$	$0.3627\text{x}10^{1}$
Eigenvalues of System Matrix	$0.9047 + j\ 0.3596 \times 10^{-12}$ $0.3676 - j\ 0.3226 \times 10^{-12}$ $0.6067 - j\ 0.3617 \times 10^{-13}$		
Corresponding s-Domain Eigenvalues	-1.001 -10.003 -4.999		

EXAMPLE 8c			
Transfer Function of The System	$\frac{4000}{(s+1)(s+5)(s+10)}$		
Sampling Per.	0.1		
DX	0.01		
DD	0.00001		
The Type of The Data	Manipulated to take the step out Rounded to four significant digits		
Realized System Matrix	0.1332×10^{-14}	0.1776×10^{-14}	0.1896
	0.1000×10^{1}	0.0	-0.1080×10^{1}
	0.1554×10^{-14}	0.1000×10^{1}	0.1866×10^{1}
Realized Distribution Matrix T	0.1000×10^{1}	-0.1110×10^{-14}	-0.2220×10^{-15}
Realized Observation Matrix	0.4537	0.2088×10^{1}	0.3628×10^{1}
Eigenvalues of System Matrix	$0.8977 - j\,0.1602 \times 10^{-14}$ $0.3316 - j\,0.1348 \times 10^{-14}$ $0.6368 + j\,0.3664 \times 10^{-14}$		
Corresponding s-Domain Eigenvalues	-1.08 -11.0 -4.51		

EXAMPLE 9a	
Transfer Function of The System	$\dfrac{80000}{(s+40)(s^2+4s+400)}$
Sampling Per.	0.1
DX	0.01
DD	0.00001
The Type of The Data	Manipulated to take the step out Rounded to eight significant digits
Realized System Matrix	$0.1942\text{x}10^{-15}$ $-0.4440\text{x}10^{-15}$ $0.1227\text{x}10^{-1}$ $0.1000\text{x}10^{1}$ $-0.4440\text{x}10^{-15}$ -0.6581 0.0 $0.1000\text{x}10^{1}$ -0.6481
Realized Distribution Matrix ^{T}	$0.1000\text{x}10^{1}$ $0.1332\text{x}10^{-14}$ $0.1332\text{x}10^{-14}$
Realized Observation Matrix	$0.4355\text{x}10^{1}$ $0.3682\text{x}10^{1}$ $-0.4645\text{x}10^{1}$
Eigenvalues of System Matrix	$-0.3332 + j\ 0.7478$ $-0.3332 - j\ 0.7478$ $0.1831 \times 10^{-1} + j\ 0.3330 \times 10^{-15}$
Corresponding s-Domain Eigenvalues	$-1.998 + j\ 19.9$ $-1.998 - j\ 19.9$ -40.0

EXAMPLE 9b	
Transfer Function of The System	$\dfrac{80000}{(s+40)(s^2+4s+400)}$
Sampling Per.	0.1
DX	0.01
DD	0.00001
The Type of The Data	Manipulated to take the step out Rounded to four significant digits
Realized System Matrix	$\begin{matrix} -0.8881\times10^{-15} & -0.4440\times10^{-15} & 0.1263\times10^{-1} \\ 0.1000\times10^{1} & 0.0 & -0.6576 \\ -0.1110\times10^{-14} & 0.1000\times10^{1} & -0.6474 \end{matrix}$
Realized Distribution Matrix T	$0.1000\times10^{1} \quad 0.6661\times10^{-15} \quad 0.2220\times10^{-15}$
Realized Observation Matrix	$0.4356\times10^{1} \quad 0.3683\times10^{1} \quad -0.4646\times10^{1}$
Eigenvalues of System Matrix	$-0.3331 + j\ 0.7478$ $-0.3331 - j\ 0.7478$ $0.1885 \times 10^{-1} + j\ 0.5551 \times 10^{-16}$
Corresponding s-Domain Eigenvalues	$-1.998 + j\ 19.9$ $-1.998 - j\ 19.9$ -39.8

EXAMPLE 10a	
Transfer Function of The System	$\dfrac{80000}{(s+40)(s+2+j10\pi)(s+2-j10\pi)}$
Sampling Per.	0.1
DX	0.00001
DD	0.00001
The Type of The Data	Manipulated to take the step out Double-Precision
Realized System Matrix	-0.2220×10^{-15} $\quad$ 0.1499×10^{-1} 0.1000×10^{1} $\quad$ -0.8004
Realized Distribution Matrix ^{T}	0.1000×10^{1} $\quad$ -0.2220×10^{-15}
Realized Observation Matrix	0.2981×10^{1} $\quad$ -0.1765×10^{1}
Eigenvalues of System Matrix	$-0.8187 + j\,0.0$ $0.1831 \times 10^{-1} + j\,0.0$
Corresponding s-Domain Eigenvalues	$-2.0 + j\,10\pi$ -40.0

EXAMPLE 10b	
Transfer Function of The System	$$\frac{80000}{(s+40)(s+2+j10\pi)(s+2-j10\pi)}$$
Sampling Per.	0.05
DX	0.00001
DD	0.00001
The Type of The Data	Manipulated to take the step out Double-Precision
Realized System Matrix	$\begin{matrix} -0.8881\times10^{-15} & -0.2775\times10^{-16} & 0.1108 \\ 0.1000\times10^{1} & 0.1387\times10^{-16} & -0.8187 \\ -0.4440\times10^{-15} & 0.1000\times10^{1} & 0.1353 \end{matrix}$
Realized Distribution Matrix ^{T}	$\begin{matrix} 0.1000\times10^{1} & 0.1387\times10^{-16} & 0.5551\times10^{-15} \end{matrix}$
Realized Observation Matrix	$\begin{matrix} 0.8901 & 0.2091\times10^{1} & -0.1333 \end{matrix}$
Eigenvalues of System Matrix	$0.8471 \times 10^{-7} + j\ 0.9048$ $0.8471 \times 10^{-7} - j\ 0.9048$ $0.1353 - j\ 0.9048 \times 10^{-14}$
Corresponding s-Domain Eigenvalues	$-2.0 + j\ 10\pi$ $-2.0 - j\ 10\pi$ -40.0

III. CONCLUSION

Ho's Algorithm provides a valuable tool for complete identification of linear-stationary systems which are completely controllable and observable. The computations required are simple and can be programmed easily, in fact for most of the computations, available library subroutines of a computer facility can be used.

For discrete systems accuracy of the measurement is the only limiting factor. For continuous systems a discrete realization seems to be more feasible as the evaluation of derivatives introduces error. The tested examples showed that a five-digits-precision data would give fairly good results for this approach.

```
C                    DIGITAL COMPUTER PROGRAMS
C
C
C
C          MAIN PROGRAM
C
C               READS THE DATA ONE ON EACH CARD WITH
C               FORMAT(D17.10). FOR DISCRETE REPRESENTATION OF
C               THE SYSTEM AT LEAST 2*M SAMPLES MUST BE
C               SUPPLIED FIRST BEING SAMPLED ONE PERIOD AFTER
C               INITIAL TIME. FOR CONTINUOUS REPRESENTATION
C               OF THE SYSTEM (EVALUATING DERIVATIVES) AT LEAST
C               8*M+1 SAMPLES MUST BE SUPPLIED FIRST BEING
C               SAMPLED AT INITIAL TIME, WHERE M IS AT LEAST
C               THE ESTIMATED UPPER LIMIT OF SYSTEM DIMENSION
C               PLUS ONE.
C
C               PROGRAM IS FOR SINGLE INPUT SINGLE OUTPUT
C               SYSTEMS,BUT CAN BE ALTERED FOR MULTIPLE
C               INPUT-OUTPUT SYSTEMS.
C               SUBROUTINES DO NOT NEED ANY CHANGE.
C
C
      IMPLICIT REAL*8(A-H,P-Z)
      COMPLEX*16 AX,VAL
      DIMENSION H(40),HO(20,20),HOT(20,20),P(20,20),Q(20,20)
     &,AA(20,20),BB(20,20),CC(20,20),AA1(20,20),HO1(20,20),
     &A(165),AX(20,20),VAL(20)
C          MM IS DIMENSION OF INPUT VECTOR
      MM=1
C          MP IS DIMENSION OF OUTPUT VECTOR
      MP=1
C          FLAG CONTROLS MODE OF ALGORITHM, FLAG=1.0 GIVES A
C          SET OF DIFFERENCE EQUATIONS, FLAG=0.0 GIVES A SET
C          OF DIFFERENTIAL EQUATIONS.
      FLAG=1.0
C          M IS THE ESTIMATED UPPER LIMIT OF THE SYSTEM ORDER
C          PLUS ONE AT LEAST
      M=5
      L=M*2
      LD=4*L+1
C          W IS SAMPLING PERIOD
      W=0.1
      WRITE(6,20) W
      IF(FLAG.GT.0.9) GO TO 91
C          READ SAMPLE VALUES
      DO 9 I=1,LD
      READ(5,111) A(I)
    9 CONTINUE
C          FIND SUCCESSIVE DERIVATIVES
      CALL SDWED2(H,W,L,A)
      GO TO 93
C          READ SAMPLE VALUES
   91 DO 12 I=1,L
      READ(5,111) A(I)
   12 CONTINUE
C          FIND PULSE RESPONSE SAMPLE VALUES
      H(1)=A(1)
      DO 92 LL=2,L
   92 H(LL)=A(LL)-A(LL-1)
   93 WRITE(6,400)
      WRITE(6,40) (H(J),J=1,L)
      DO 15 K=1,M
C          BUILT HO AND HOT MATRICES
      DO 14 I=1,K
      DO 14 J=1,K
      NL=I+J-1
      HOT(I,J)=H(NL+1)
      HO1(I,J)=H(NL)
   14 HO(I,J)=H(NL)
      K1=K*MM
      DX=0.001
```

```
      DD=0.00001
C         FIND THE RANK OF HO
      CALL RANK(K,K1,HO1,DX,IC)
      WRITE(6,50) IC,K
C         FIND P AND Q MATRICES
      CALL PHQQ(K,K1,HO,DD,P,Q,IC)
C         MULTIPLY P AND HOT
      CALL MULT(K,P,HOT,K,K1,AA1)
      WRITE(6,60)
C         MULTIPLY THE RESULT WITH Q
      CALL MULT(K1,AA1,Q,IC,IC,AA)
C         WRITE SYSTEM MATRIX
      DO 4 IA=1,IC
    4 WRITE(6,10) (AA(IA,JA),JA=1,IC)
      WRITE(6,500)
      WRITE(6,70)
C         MULTIPLY P AND HO
      CALL MULT(K,P,HO,IC,MM,BB)
C         WRITE DISTRIBUTION MATRIX
      DO 2 IA=1,IC
    2 WRITE(6,10) (BB(IA,JA),JA=1,MM)
      WRITE(6,500)
      WRITE(6,80)
C         MULTIPLY HO AND Q
      CALL MULT(K1,HO,Q,MP,IC,CC)
C         WRITE OBSERVATION MATRIX
      DO 3 IA=1,MP
    3 WRITE(6,10) (CC(IA,JA),JA=1,IC)
C         FIND THE EIGENVALUES OF SYSTEM MATRIX. DALMAT IS A
C         LIBRARY SUBROUTINE (N.P.G.S. COMPUTER FACILITY)
C         WHICH FINDS THE COMPLEX EIGENVALUES OF A COMPLEX
C         MATRIX
      DO 5 IA=1,IC
      DO 5 JA=1,IC
    5 AX(IA,JA)=AA(IA,JA)
      CALL DALMAT(AX,VAL,IC,20,NCAL)
      WRITE(6,90) (VAL(IA),IA=1,IC)
   15 CONTINUE
   10 FORMAT(/,7D18.10)
   20 FORMAT(///T20,'SAMPLING PERIOD=',D16.8,1X,'SECONDS',/)
   40 FORMAT(7D18.10)
   50 FORMAT(//T4,'RANK HO=',I3,5X,'DIMENSION HO=',I3,///)
   60 FORMAT(T4,'A MATRIX',//)
   70 FORMAT(T4,'B MATRIX',//)
   80 FORMAT(T4,'C MATRIX',//)
   90 FORMAT(//,10X,'EIGENVALUE=',2D18.10)
  111 FORMAT(D17.10)
  400 FORMAT(//T20,'H VALUES',/)
  500 FORMAT(/////)
      RETURN
      END
C
C
C         SUBROUTINE SDWFD2
C
C         PURPOSE:
C              SUBROUTINE SDWFD2 FINDS THE SUCCESSIVE
C              DERIVATIVES OF A FUNCTION EVALUATED AT ZERO
C              VALUE OF THE INDEPENDENT VARIABLE USING
C              NEWTON FORWARD DIFFERENCES.
C
C         DESCRIPTION OF PARAMETERS:
C              A--ORDINATE VALUES OF FUNCTION
C                 SAMPLED WITH PERIOD W
C              ID--HIGHEST DERIVATIVE TO BE FOUND
C              W--SAMPLING PERIOD
C              H--CONTAINS SUCCESSIVE DERIVATIVES
C              A,ID,W ARE INPUTS,H IS OUTPUT
C
C         NOTE:
C              FOR IDTH DERIVATIVE 4*ID+1 SAMPLES ARE NEEDED
C
```

```
C
      SUBROUTINE SDWFD2(H,W,ID,A)
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION H(40),A(165),B(165),C(165),D(165),E(165)
C         INITIALIZE THE INDECES
      K=4*ID
      L=4*ID-1
      M=4*ID-2
      N=4*ID-3
      DO 7 J=1,ID
C         FIND THE FIRST DIFFERENCES
      DO 2 I=1,K
    2 B(I)=A(I+1)-A(I)
C         FIND THE SECOND DIFFERENCES
      DO 3 I=1,L
    3 C(I)=B(I+1)-B(I)
C         FIND THE THIRD DIFFERENCES
      DO 4 I=1,M
    4 D(I)=C(I+1)-C(I)
C         FIND THE FOURTH DIFFERENCES
      DO 5 I=1,N
    5 E(I)=D(I+1)-D(I)
C         CALCULATE THE JTH DERIVATIVE AND PUT IT TO SAMPLE
C         ARRAY TO FIND THE NEXT DERIVATIVE
      DO 6 I=1,N
    6 A(I)=(B(I)-0.5*C(I)+(1.0/3.0)*D(I)-0.25*E(I))/W
C         PUT THE DERIVATIVE EVALUATED AT
C         ZERO TO DERIVATIVE ARRAY
      H(J)=A(1)
      K=K-4
      L=L-4
      M=M-4
    7 N=N-4
      RETURN
      END
C
C
C         SUBROUTINE RANK
C
C         PURPOSE:
C              SUBROUTINE RANK FINDS THE RANK OF REAL MATRICES
C              WITH DIMENSIONS UPTO (20,20)
C
C         DESCRIPTION OF PARAMETERS:
C              A--INPUT MATRIX DIMENSIONED REAL*8 A(20,20)
C              M--ACTUAL ROW DIMENSION OF A
C              N--ACTUAL COLUMN DIMENSION OF A
C              IC--AT THE END IC IS THE RANK OF A
C              DD--A SUITABLE SMALL POSITIVE NUMBER,DURING
C              OPERATION ANY NUMBER WHICH HAS AN ABSOLUTE
C              VALUE LESS THAN DD IS MADE EQUAL TO ZERO
C              A,M,N,DD ARE INPUTS IC IS OUTPUT
C
C         NOTE:
C              THE A MATRIX IS DESTROYED
C
C
      SUBROUTINE RANK(M,N,A,DD,IC)
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION A(20,20)
C         IC INITIALIZED TO ZERO AT THE START
      IC=0
      DO 14 IR=1,M
C         EACH TIME BIG IS INITIALIZED AS
C         FIRST ELEMENT OF IRTH ROW
      BIG=DABS(A(IR,1))
      I=IR
      J=1
C         SEARCH THE MATRIX FOR LARGEST ABSOLUTE VALUE
C         ELEMENT BELOW THE IRTH ROW (INCLUDING)
      DO 5 L=1,N
      DO 5 K=IR,M
```

```
      IF(BIG.GE.DABS(A(K,L))) GO TO 5
C         BIG IS THE LARGEST ABSOLUTE VALUE
C         ELEMENT DURING SEARCH
      BIG=DABS(A(K,L))
C         MEMORIZE THE ROW AND COLUMN OF BIG
      I=K
      J=L
    5 CONTINUE
C         IF BIG IS IN IRTH ROW GO TO 8,OTHERWISE INTERCHANGE
C         THE ROW OF BIG AND THE IRTH ROW
      IF(I.EQ.IR) GO TO 8
      DO 7 L=1,N
      TEMPA=A(I,L)
      A(I,L)=A(IR,L)
    7 A(IR,L)=TEMPA
    8 IF(BIG.LT.DD) GO TO 14
C         IF BIG IS NOT ZERO INCREASE RANK ONE
      IC=IC+1
C         REDUCE ALL ELEMENTS ABOVE AND BELOW OF BIG TO ZERO
      DO 12 K=1,M
      IF(K.EQ.IR) GO TO 12
      IF(A(K,J).EQ.0.0) GO TO 12
      G=A(K,J)/A(IR,J)
      DO 11 L=1,N
      IF(L.EQ.J) GO TO 13
      A(K,L)=A(K,L)-G*A(IR,L)
      IF(DABS(A(K,L)).LT.DD) A(K,L)=0.0
   13 A(K,J)=0.0
   11 CONTINUE
   12 CONTINUE
   14 CONTINUE
      RETURN
      END
C
C
C         SUBROUTINE PHOQ
C
C         PURPOSE:
C              SUBROUTINE PHOQ FINDS THE ELEMENTARY
C              TRANSFORMATION MATRICES P AND Q WHICH
C              PUT THE A MATRIX INTO ITS NORMAL FORM.
C
C         DESCRIPTION OF PARAMETERS:
C              A--THE MATRIX ITS NORMAL FORM WILL BE FOUND
C              M--ACTUAL ROW DIMENSION OF A
C              N--ACTUAL COLUMN DIMENSION OF A
C              IC--RANK OF A
C              P--ELEMENTARY ROW TRANSFORMATION MATRIX
C              Q--ELEMENTARY COLUMN TRANSFORMATION MATRIX
C              DD--A SUITABLE SMALL POSITIVE NUMBER,DURING
C              OPERATION ANY NUMBER WHICH HAS AN ABSOLUTE
C              VALUE LESS THAN DD IS MADE EQUAL TO ZERO
C              M,N,A,DD,IC ARE INPUTS,P AND Q ARE OUTPUTS
C
C         NOTE:
C              A,P AND Q MATRICES ARE FIRST ARRANGED IN THE X
C              ARRAY SUCH THAT Q IS ON TOP AND P ON LEFT OF A.
C              P AND Q ARE IDENTITY MATRICES AT THE BEGINING.
C
C
      SUBROUTINE PHOQ(M,N,A,DD,P,Q,IC)
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION A(20,20),P(20,20),Q(20,20),X(40,40)
C         FIND THE DIMENSION OF X
      K1=M+N
C         INITIALIZE X TO ZERO
      DO 1 I=1,K1
      DO 1 J=1,K1
    1 X(I,J)=0.0
C         INITIALIZE P PART OF X TO IDENTITY
      K2=N+1
      DO 2 I=K2,K1
```

```
      DO 2 J=1,M
      IF(J.NE.(I-N)) GO TO 2
      X(I,J)=1.0
    2 CONTINUE
C         INITIALIZE Q PART OF X TO IDENTITY
      K3=M+1
      DO 3 I=1,N
      DO 3 J=K3,K1
      IF(I.NE.(J-M)) GO TO 3
      X(I,J)=1.0
    3 CONTINUE
C         PUT THE A MATRIX INTO ITS PLACE AT X ARRAY
      DO 4 I=1,M
      DO 4 J=1,N
    4 X((I+N),(J+M))=A(I,J)
C         K3 IS THE COLUMN OF X WHERE FIRST COLUMN OF THE A
C         AND Q MATRICES ARE
      DO 21 J=K3,K1
C         BIG IS THE LARGEST ELEMENT IN COLUMN
C         SEARCH INITIALIZED TO ZERO
      BIG=0.0
C         AFTER IC COLUMNS OF A IS PUT TO ITS NORMAL FORM BY
C         ROW TRANSFORMATIONS GO TO 22
    5 IF((J-M).GT.IC) GO TO 22
      K4=K2-K3+J
      DO 11 I=K4,K1
C         SEARCH THE COLUMN OF A FOR LARGEST ELEMENT BELOW K4
C         MEMORIZE ROW AND COLUMN OF LARGEST ELEMENT
      IF(BIG.GE.DABS(X(I,J))) GO TO 11
      BIG=DABS(X(I,J))
      IROW=I
      JCOL=J
   11 CONTINUE
      IF(BIG.NE.0.0) GO TO 14
C         IF ALL ELEMENTS OF COLUMN ARE ZERO REMOVE THIS
C         COLUMN SHIFT ALL COLUMNS AT RIGHT OF THIS ONE PLACE
C         TO THE LEFT AND PUT REMOVED COLUMN TO LAST COLUMN
      DO 13 II=1,K1
      TEMP=X(II,J)
      KK1=K1-1
      DO 12 JJ=J,KK1
   12 X(II,JJ)=X(II,(JJ+1))
      X(II,K1)=TEMP
   13 CONTINUE
      GO TO 5
   14 IF(IROW.EQ.K4) GO TO 16
C         IF BIG IS NOT IN APPROPRIATE ROW CHANGE ROWS
      DO 15 J1=1,K1
      TEMP=X(K4,J1)
      X(K4,J1)=X(IROW,J1)
      X(IROW,J1)=TEMP
   15 CONTINUE
C         DIVIDE ROW OF BIG BY BIG
   16 TEMP=X(K4,J)
      DO 17 J1=1,K1
   17 X(K4,J1)=X(K4,J1)/TEMP
      DO 19 I=K2,K1
      IF(I.EQ.K4) GO TO 19
      IF(X(I,J).EQ.0.0) GO TO 19
      G=X(I,J)
      DO 18 J1=1,K1
      X(I,J1)=X(I,J1)-G*X(K4,J1)
      IF(DABS(X(I,J1)).LT.DD) X(I,J1)=0.0
   18 CONTINUE
   19 CONTINUE
   21 CONTINUE
C         AT THIS POINT ALL ROWS OF THE A BELOW ICTH SHOULD
C         BE ZERO, IF RANK EQUAL TO COLUMN DIMENSION OF
C         THE A,NORMAL FORM IS OBTAINED, OTHERWISE BY COLUMN
C         TRANSFORMATIONS FINISH THE ALGORITHM
   22 K5=K3+IC
      K6=N+IC
```

```
      L1=K3
      IF(IC.EQ.N) GO TO 41
      DO 33 L=K2,K6
      DO 32 J=K5,K1
      IF(X(L,J).EQ.0.0) GO TO 32
      G=X(L,J)
      DO 31 I=1,K1
      X(I,J)=X(I,J)-G*X(I,L1)
      IF(DABS(X(I,J)).LT.DD) X(I,J)=0.0
   31 CONTINUE
   32 CONTINUE
      L1=L1+1
   33 CONTINUE
C        PUT P TO ITS ARRAY
   41 DO 23 I=1,M
      DO 23 J=1,M
   23 P(I,J)=X((I+N),J)
C        PUT Q TO ITS ARRAY
      DO 24 I=1,N
      DO 24 J=1,N
   24 Q(I,J)=X(I,(J+M))
      RETURN
      END
C
C
C        SUBROUTINE MULT
C
C        PURPOSE:
C             SUBROUTINE MULT FINDS THE (IC,IM) NORTHWEST
C             CORNER OF C=AB MATRIX MULTIPLICATION
C
C        DESCRIPTION OF PARAMETERS:
C             A--MULTIPLICAND MATRIX
C             B--MULTIPLIER MATRIX
C             C--RESULTANT MATRIX
C             L--ACTUAL COLUMN DIMENSION OF THE A
C             IC--ROW DIMENSION OF DESIRED NORTHWEST CORNER
C             IM--COLUMN DIMENSION OF DESIRED N.W. CORNER
C             A,B,L,IC,IM ARE INPUTS,C IS OUTPUT
C
C
      SUBROUTINE MULT(L,A,B,IC,IM,C)
      IMPLICIT REAL*8(A-H,P-Z)
      DIMENSION A(20,20),B(20,20),C(20,20)
      DO 3 I=1,IC
      DO 2 J=1,IM
      SUM=0.0
      DO 1 K=1,L
    1 SUM=SUM+A(I,K)*B(K,J)
    2 C(I,J)=SUM
    3 CONTINUE
      RETURN
      END
```

BIBLIOGRAPHY

1. B. L. Ho and R. E. Kalman, "Effective Construction of Linear State-Variable Models from Input/Output Functions", Sonderdruck aus "Regelungstechnik", 14. Jahrg., Heft 12, 1966, Seite 545-548.

2. R. E. Kalman, "Mathematical Description of Linear Dynamical Systems", J.S.I.A.M. Control, Ser. A, Vol.1, No. 2, 1963.

3. Elmer G. Gilbert, "Controllability and Observability in Multivariable Control Systems", J.S.I.A.M. Control, Ser. A, Vol. 2, No. 1, 1963.

4. Paul Zeiger, "Lecture Notes on Algebraic Theory of Linear Systems".

5. Robert C. K. Lee, "Optimal Estimation, Identification, and Control", Research Monograph No. 28, The M.I.T. Press.

6. R. E. Kalman, "On the General Theory of Control Systems", Proc. 1st International Congress on Automatic Control, Moscow, 1960.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	R. E. Kalman Department of Operation Research Stanford University Stanford, California 94305	1
4.	Asst. Professor D. E. Kirk Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Asst. Professor J. S. Demetry Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	Assoc. Professor H. A. Titus Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	Professor G. J. Thaler Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
8.	Dr. R. C. Dorf, Dean School of Engineering Ohio University Athens, Ohio 45701	1
9.	LT I. Eldem Günaydin Mah. Istiklal Cad Ar-iş Ap. Kat.3 Bandirma, TURKEY	4
10.	Deniz Kuvvetleri Komutanligi Ankara, TURKEY	1

		No. Copies
11.	Deniz Harb Okulu Komutanligi Heybeliada - Istanbul - TURKEY	1
12.	Middle-East Technical University Ankara, TURKEY	1
13.	Istanbul Teknik Universitesi Taskisla - Istanbul, TURKEY	1
14.	Karadeniz Teknik Universitesi Trabzon, TURKEY	1

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Identification of Plant Dynamics Using Ho's Algorithm

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Master's Thesis; June 1969

5. AUTHOR(S) *(First name, middle initial, last name)*

Ilker ELDEM

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1969	66	6

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

Ho's Algorithm is reviewed and demonstrated with analytic examples. A digital computer program is developed to implement the algorithm for single input, single output systems and used to identify linear continuous and stationary systems which are driven with a unit step as the test input. Discrete realization of the continuous systems is obtained using the measured output samples, to a step input, directly in the algorithm.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Identification						
Algorithm						
Linear Systems						